CLAIMS

1. A method of transmitting data packets between a first computer and a second computer, wherein the first computer and the second computer are linked via a plurality of separate transmission paths, the method comprising the steps of:

(1) assigning a weight value to each of the plurality of transmission paths, wherein each respective weight value represents the relative number of packets that a respective transmission path will transmit;

(2) for each data packet that is to be transmitted from the first computer to the second computer, selecting one of the plurality of transmission paths on the basis of each respective transmission path's assigned weight value;

(3) measuring the transmission quality for each of the plurality of transmission paths; and

(4) adjusting downwardly to a non-zero value the assigned weight value for a transmission path for which the transmission quality has declined.

2. The method of claim 1, wherein step (4) comprises the step of gradually decreasing over time the assigned weight value in relation to weight values assigned to the remaining transmission paths.

3. The method of claim 2, wherein step (4) comprises the step of gradually decreasing the assigned weight value according to an incrementally decreasing function.

4. The method of claim 2, wherein step (4) comprises the step of gradually decreasing the assigned weight value according to an exponentially decaying function.

5. The method of claim 1, wherein step (3) comprises the step of determining that one or more packets transmitted to the second computer was not acknowledged by the second computer.

6. The method of claim 1, wherein step (3) comprises the step of evaluating the contents of a synchronization packet that maintains synchronization with a moving window of valid values.

7. The method of claim 1, further comprising the step of inserting into each data packet a source and destination IP address pair that is selected according to a pseudo-random sequence.

8. The method of claim 1, wherein step (4) comprises the step of adjusting downwardly the assigned weight value for a transmission path only if the transmission quality has declined below a predetermined threshold.

9. The method of claim 1, further comprising the step of adjusting upwardly the assigned weight value that was adjusted in step (4) if it is later determined that the transmission quality has improved.

10. The method of claim 1, further comprising the step of adjusting upwardly the weight values of the remaining transmission links in an amount that compensates for the downwardly adjusted weight value.

11. The method of claim 10, wherein the step of adjusting upwardly comprises the step of equally distributing the amount that was downwardly adjusted across the remaining transmission links.

12. The method of claim 1, further comprising the step of adjusting downwardly to zero the assigned weight value for any transmission link whose quality has degraded below a preset threshold.

13. The method of claim 1, wherein steps (2) through (4) are repeated periodically.

14. A first computer that transmits data packets to a second computer over a plurality of separate transmission paths, wherein the first computer performs the steps of:

(1) assigning a weight value to each of the plurality of transmission paths, wherein each respective weight value represents the relative number of packets that a respective transmission path will transmit;

(2) for each data packet that is to be transmitted to the second computer, selecting one of the plurality of transmission paths on the basis of each respective transmission path's assigned weight value;

(3) measuring the transmission quality for each of the plurality of transmission paths; and

(4) adjusting downwardly to a non-zero value the assigned weight value for a transmission path for which the transmission quality has declined.

15. The first computer of claim 14, wherein the first computer gradually decreases over time the assigned weight value in relation to weight values assigned to the remaining transmission paths.

16. The first computer of claim 15, wherein the first computer gradually decreases the assigned weight value according to an incrementally decreasing function.

17. The first computer of claim 15, wherein the first computer gradually decreases the assigned weight value according to an exponentially decaying function.

18. The first computer of claim 14, wherein the first computer measures the transmission quality by determining that one or more packets transmitted to the second computer was not acknowledged by the second computer.

19. The first computer of claim 14, wherein the first computer measures the transmission quality by evaluating the contents of a synchronization packet that maintains synchronization with a moving window of valid values.

20. The first computer of claim 14, wherein the first computer inserts into each data packet a source and destination IP address pair that is selected according to a pseudo-random sequence.

21. The first computer of claim 14, wherein the first computer adjusts downwardly the assigned weight value for any transmission path only if the transmission quality has declined below a predetermined threshold.

22. The first computer of claim 14, wherein the first computer adjusts upwardly the assigned weight value that was adjusted in step (4) if it is later determined that the transmission quality has improved.

23. The first computer of claim 14, wherein the first computer adjusts upwardly the weight values of the remaining transmission links in an amount that compensates for the downwardly adjusted weight value.

24. The first computer of claim 23, wherein the first computer upwardly adjusts probabilities across the remaining transmission links in an amount equal to the downwardly adjusted weight value.

25. The first computer of claim 14, wherein the first computer adjusts downwardly to zero the assigned weight value for any transmission link whose quality has degraded below a preset threshold.

26. The first computer of claim 14, wherein the first computer repeats steps (2) through (4) periodically.

27. A system comprising the first computer of claim 14 and a second computer constructed in accordance with the first computer of claim 14.

28. A method of transparently creating a virtual private network (VPN) between a client computer and a target computer, comprising the steps of:

(1) generating from the client computer a Domain Name Service (DNS) request that requests an IP address corresponding to a domain name associated with the target computer;

(2) determining whether the DNS request transmitted in step (1) is requesting access to a secure web site; and

(3) in response to determining that the DNS request in step (2) is requesting access to a secure target web site, automatically initiating the VPN between the client computer and the target computer.

29. The method of claim 28, wherein steps (2) and (3) are performed at a DNS server separate from the client computer.

30. The method of claim 28, further comprising the step of:

(4) in response to determining that the DNS request in step (2) is not requesting access to a secure target web site, resolving the IP address for the domain name and returning the IP address to the client computer.

31. The method of claim 28, wherein step (3) comprises the step of, prior to automatically initiating the VPN between the client computer and the target computer, determining whether the client computer is authorized to establish a VPN with the target computer and, if not so authorized, returning an error from the DNS request.

32. The method of claim 28, wherein step (3) comprises the step of, prior to automatically initiating the VPN between the client computer and the target computer,

determining whether the client computer is authorized to resolve addresses of non secure target computers and, if not so authorized, returning an error from the DNS request.

33. The method of claim 28, wherein step (3) comprises the step of establishing the VPN by creating an IP address hopping scheme between the client computer and the target computer.

34. The method of claim 28, wherein step (3) comprises the step of using a gatekeeper computer that allocates VPN resources for communicating between the client computer and the target computer.

35. The method of claim 28, wherein step (2) is performed in a DNS proxy server that passes through the request to a DNS server if it is determined in step (3) that access is not being requested to a secure target web site.

36. The method of claim 32, wherein step (3) comprises the step of transmitting a message to the client computer to determine whether the client computer is authorized to establish the VPN target computer.

37. A system that transparently creates a virtual private network (VPN) between a client computer and a secure target computer, comprising:

a DNS proxy server that receives a request from the client computer to look up an IP address for a domain name, wherein the DNS proxy server returns the IP address for the requested domain name if it is determined that access to a non-secure web site has been requested, and wherein the DNS proxy server generates a request to create the VPN between the client computer and the secure target computer if it is determined that access to a secure web site has been requested; and

a gatekeeper computer that allocates resources for the VPN between the client computer and the secure web computer in response to the request by the DNS proxy server.

38. The system of claim 37, wherein the gatekeeper computer creates the VPN by establishing an IP address hopping regime that is used to pseudorandomly change IP addresses in packets transmitted between the client computer and the secure target computer.

39. The system of claim 37, wherein the gatekeeper computer determines whether the client computer has sufficient security privileges to create the VPN and, if the client computer lacks sufficient security privileges, rejecting the request to create the VPN.

40. A method of preventing data packets received from a high bandwidth link from flooding a low bandwidth link, comprising the steps of:

(1) receiving data packets from the high bandwidth link that are ostensibly addressed to a computer residing on the low-bandwidth link;

(2) for each data packet, determining whether the data packet is validly addressed to the computer on the low-bandwidth link;

(3) in response to determining that the data packet is not validly addressed to the computer on the low-bandwidth link, rejecting the data packet; and

(4) in response to determining that the data packet is validly addressed to the computer on the low-bandwidth link, forwarding the data packet to the computer over the low-bandwidth link.

41. The method of claim 40, wherein step (3) comprises the step of comparing a value in a header of each data packet to a set of valid values maintained for the computer on the low-bandwidth link.

42. The method of claim 41, wherein step (3) comprises the step of comparing a value in a header of each data packet to a moving window of valid values.

43. The method of claim 42, wherein step (3) comprises the step of comparing the IP address in the header of each data packet to a moving window of valid IP addresses, wherein the moving window is also maintained by the computer on the low-bandwidth link.

44. The method of claim 40, wherein step (3) comprises the step of reducing a priority level of the packet in relation to other data packets, wherein the priority level determines whether a particular data packet will be transmitted before another data packet having a different priority level.

45. The method of claim 40, wherein step (3) comprises the step of performing a cryptographic check on each data packet to determine whether each data packet is validly addressed.

46. The method of claim 40, wherein step (3) comprises the step of receiving a message from the computer on the low-bandwidth link to stop accepting messages having a particular characteristic.

47. The method of claim 46, wherein step (3) comprises the step of receiving a message from the computer on the low-bandwidth link to stop accepting messages addressed to a particular IP address.

48. The method of claim 40, wherein step (3) comprises the step of determining that a packet transmission rate has been exceeded for a given packet parameter.

49. The method of claim 48, wherein step (3) comprises the step of determining that a packet transmission rate has been exceeded for a given IP destination address.

50. In a system having a low bandwidth data link, a first computer coupled to the low bandwidth data link, and a high bandwidth data link, an improvement comprising:

a second computer coupled between the low bandwidth data link and the high bandwidth data link, wherein the second computer receives data packets from the high bandwidth data link and, if they are addressed to the first computer, routes them to the first computer over the low bandwidth data link,

wherein the second computer prevents invalid data packets ostensibly addressed to the first computer from being transmitted over the low bandwidth data link.

51. The system of claim 50, wherein the second computer prevents invalid data packets from being transmitted over the low bandwidth data link by comparing a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for the first computer.

52. The system of claim 50, wherein the second computer compares an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses.

53. The system of claim 52, wherein the second computer compares the IP address in the header of each data packet to a moving window of valid IP addresses, wherein the moving window is also maintained by the first computer.

54. The system of claim 50, wherein the second computer reduces a priority level of a data packet in relation to other data packets, wherein the priority level determines whether a particular data packet will be transmitted before another data packet having a different priority level.

55. The system of claim 50, wherein the second computer performs a cryptographic check on each data packet to determine whether each data packet is validly addressed.

56. The system of claim 50, wherein the second computer receives a message from the first computer that causes the second computer to stop accepting messages having a particular characteristic.

57. The system of claim 56, wherein the second computer receiving a message from the first computer to stop accepting messages addressed to a particular IP address.

58. The system of claim 50, wherein the second computer rejects invalid packets by determining that a packet transmission rate has been exceeded for a given packet parameter.

59. The system of claim 58, wherein the second computer determines that a packet transmission rate has been exceeded for a given IP destination address.

60. In a system comprising a first computer that transmits data packets to a second computer over a network according to a scheme by which at least one field in a series of data packets is periodically changed according to a sequence known by the first and second computers, and wherein the second computer periodically receives a synchronization request from the first computer to maintain synchronization of the sequence between the first and second computers, a method comprising the steps of:

(1) receiving at the first computer the synchronization request from the second computer;

(2) determining whether the synchronization request was received in less than a predetermined interval;

(3) in response to determining that the synchronization request was received in less than the predetermined interval, ignoring the synchronization request; and

(4) in response to determining that the synchronization request was not received in less than the predetermined interval, providing the synchronization response to the first computer.

61. The method of claim 60, wherein step (3) comprises the step of delaying the acceptance of a SYNC_REQ for W/R seconds, where W is the number of data packets between synchronization requests according to an agreed schedule, and R is the agreed rate at which synchronization requests should be received according to the agreed schedule.

62. The method of claim 60, further comprising the step of determining whether the synchronization request is a duplicate of a previously received synchronization request and, if it is a duplicate, discarding it.

63. The method of claim 60, wherein step (4) comprises the step of providing a response that includes a new checkpoint for synchronizing a window in a hopping table.

64. A computer that receives data packets from a second computer over a network according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence, wherein the second computer periodically transmits a synchronization request to maintain synchronization of the sequence, wherein the computer performs the steps of:

(1) receiving the synchronization request from the second computer;

(2) determining whether the synchronization request was received in less than a predetermined interval;

(3) in response to determining that the synchronization request was received in less than a predetermined interval ignoring the synchronization request; and

(4) in response to determining that the synchronization request was not received in less than a predetermined interval, providing the response to the first computer.

65. The computer of claim 64, wherein the computer delays the acceptance of a SYNC_REQ in step (3) for W/R seconds, where W is the number of data packets between synchronization requests according to an agreed schedule, and R is the agreed rate at which synchronization requests should be received according to the agreed schedule.

66. The computer of claim 64, wherein the computer further performs the step of determining whether the synchronization request is a duplicate of a previously received synchronization request and, if it is a duplicate, discarding it.

13
67. A method of establishing communication between one of a plurality of client computers and a central computer that maintains a plurality of authentication tables each corresponding to one of the client computers, the method comprising the steps of:

(1) in the central computer, receiving from one of the plurality of client computers a request to establish a connection;

(2) authenticating, with reference to one of the plurality of authentication tables, that the request received in step (1) is from an authorized client;

(3) responsive to a determination that the request is from an authorized client, allocating resources to establish a virtual private link between the client and a second computer; and

(4) communicating between the authorized client and the second computer using the virtual private link.

14 13
68. The method of claim 67, wherein step (4) comprises the step of communicating according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence.

15 14
69. The method of claim 68, wherein step (4) comprises the step of comparing an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses maintained in a table in the second computer.

16 15
70. The method of claim 69, wherein step (4) comprises the step of comparing the IP address in the header of each data packet to a moving window of valid IP addresses, and rejecting data packets having IP addresses that do not fall within the moving window.

17 13
71. The method of claim 67, wherein step (2) comprises the step of using a checkpoint data structure that maintains synchronization of a periodically changing parameter known by the central computer and the client computer to authenticate the client.

Add R1

Add C1

74